2014 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)
July 6-11, 2014, Beijing, China

Possibilistic Projected Categorical Clustering via Cluster Cores

Stephen G. Matthews and Trevor P. Martin

***Abstract*— Projected clustering discovers clusters in subsets of locally relevant attributes. There is uncertainty and imprecision about how groups of categorical values are learnt from data for projected clustering and also the data itself. A method is presented for learning discrete possibility distributions of categorical values from data for projected clustering in order to model uncertainty and imprecision. Empirical results show that fewer, more accurate, more compact, and new clusters can be discovered by using possibility distributions of categorical values when compared to an existing method based on Boolean memberships. This potentially allows for new relationships to be identified from data.**

I. Introduction

CLUSTERING is a task for exploratory data analysis that uses unsupervised learning to sort unlabelled/uncategorised data objects into groups according to attributes of the data objects. Traditional approaches use all attributes when measuring the similarity between data objects. However, using all attributes has been shown to be less effective on high-dimensional data [1] because there is a lack of contrast in distance between data objects as the number of dimensions increases. This is referred to as the "curse of dimensionality" [2].

For answering specific questions, scientific design can select only those attributes considered to be relevant to the question. An increasing volume of data with many attributes, i.e., high dimensional, is being recorded nowadays, and there is less focus on specific questions. Alleviating the "curse of dimensionality" in clustering by removing irrelevant attributes is challenging when there is no prior knowledge of what is relevant. Global dimensionality reduction methods, such as feature selection and feature transformation (e.g., principal component analysis), do not overcome the problem because relevant attributes of one cluster may be irrelevant for another cluster.

Projected clustering overcomes this problem by finding clusters in subsets of attributes, also known as subspaces. Consider $D = (x_{ij})$ to be a dataset of $i = 1, \ldots, n$ data objects and $j = 1, \ldots, d$ attributes, and $A = \{a_1, \ldots, a_d\}$ to be the set of attributes in D. Each attribute a_j is associated with a discrete domain $\mathrm{dom}(a_j), j = 1, \ldots, d$. A projected cluster is defined as a pair (X_i, Y_i), where X_i is a subset of data objects in D, Y_i is a subset of dimensions in D such that the points in X_i project along each attribute in Y_i ("relevant" attributes) onto a small range of values compared to the whole dataset, and the points in X_i are uniformly distributed along every other attribute not in Y_i ("irrelevant" attributes). The aim of projected clustering is to find k projected clusters (X_i, Y_i) where $i = 1, \ldots, k$, and to find a set of outliers O such that the dataset D can be partitioned into clusters and outliers, i.e., $\{X_1, \ldots, X_k, O\}$. Projected clustering finds subsets of attributes that are *local* to data objects rather than using *global* dimensionality reduction.

The Projected Clustering via Cluster Cores (P3C) algorithm [3] handles all-numeric data, all-categorical data, or mixed data (by discretising numeric attributes and applying the categorical approach). For a numerical attribute, an interval $S = [v_l, v_u]$ for attribute a_j is defined for all real values x such that $v_l \leq x \leq v_u$. For a categorical attribute, an interval is defined as $S = \{x_{i_1 j}, \ldots, x_{i_h j}\} \subseteq \mathrm{dom}(a_j)$ for h values of an attribute a_j. In P3C, intervals of attribute values are used to search for projected clusters, which is similar to set-valued rules [4]. P3C's notion of "adjacency" relationships between values of numerical attributes comes from the natural order of numerical values. However, there is no natural order in categorical data, so P3C's "adjacency" relationship for categorical attributes is based on a transitive relation where if a is related to b and b is related to c then a is also related to c.

Consider the example dataset in Table I. There are 5 data objects representing people with 3 categorical attributes. Beijing, Bristol, and London are "adjacent" because they have another attribute in common, hip hop. Bristol and London are "adjacent" because they have multiple, other attributes in common, hip hop and nurse. P3C requires only one other attribute in common, so the additional information (two other attributes in common) is not used for the "adjacency" relationship between Bristol and London. In this example, the additional information indicates a stronger "adjacency" relationship between Bristol and London (with two common attributes) than between Beijing and Bristol (with one common attribute), which is not used in P3C.

TABLE I: Example data from customer profiling to demonstrate P3C's "adjacency" relationship

	municipality	music	job
1	Beijing	hip hop	chef
2	Bristol	hip hop	nurse
3	Bristol	hip hop	nurse
4	London	hip hop	nurse
5	Ipswich	reggae	researcher

We capture the additional information from the stronger "adjacency" relationship, and its uncertainty and imprecision, by modelling the "adjacency" relationship with possibility

Stephen G. Matthews and Trevor P. Martin are with the Intelligent Systems Laboratory, Department of Engineering Mathematics, University of Bristol, Bristol BS8 1UB, UK (email: {stephen.matthews, trevor.martin}@bristol.ac.uk).

This work was supported by the Technology Strategy Board and BT plc. under grant 16707-120182.

978-1-4799-2072-3/14/$31.00 ©2014 IEEE

distributions. The additional information builds upon P3C by providing a richer representation of the "adjacency" relationship. We explore a method for learning discrete possibility distributions of categorical values using P3C's "adjacency" relationship and introduce the Possibilistic Projected Categorical Clustering via Cluster Cores (2P4C) algorithm. The 2P4C algorithm extends P3C by incorporating possibility distributions.

The outline of the paper is as follows: Section II discusses related work, Section III introduces preliminary requisites such as definitions and P3C, Section IV introduces 2P4C, Section V presents experimentation and results, and Section VI draws conclusions.

II. Related Work

Different types and categorisations of "subspace" clustering are defined first in order to explain projected clustering and related clustering methods. Various types of clustering algorithm are reviewed for subspace and projected clustering on categorical data and fuzzy and possibilistic clustering.

The terms "subspace" and "projected" have been used interchangeably with some confusion in the literature [5], and "subspace" clustering is often used as an umbrella term. According to the type of problem that clustering tackles, there are several types of "subspace" clustering [5].

Subspace clustering finds *all* clusters of objects in *all* subspaces of a dataset according to the definition of a cluster. CLIQUE [6] is a subspace clustering algorithm for numeric values that performs an Apriori-like search, and its effectiveness is determined by the granularity and positioning of the grid used for discretising numeric values.

Projected clustering finds a unique assignment of each point to exactly one subspace cluster or noise. PROCLUS [7] is a projected k-medoid-like clustering algorithm with a randomised approach that results in different clusterings.

Soft-projected clustering allows the soft assignment of points to clusters (as opposed to hard assignment in projected clustering), so clusters can overlap. LAC [8] is a soft-projected clustering algorithm that has parameters for the number of clusters k and the incentive to cluster on more features h, and assigns weights to features according to the local variance of data along each dimension.

P3C is a projected clustering algorithm that has also been categorised as a hybrid clustering algorithm [5], because P3C does not find all clusters in all subspaces and P3C can operate with hard/soft assignment of points to clusters. P3C uses one parameter, which affects the sensitivity of finding projected clusters, and finds maximal patterns that can be understood as a more compact set of results when compared to other Apriori-like algorithms.

It should be noted that projected clustering was proposed independently and simultaneously to Probabilistic Latent Semantic Indexing (PLSI) [9][10], which has similar aims of clustering and dimensionality reduction but on text.

Subspace clustering can be categorised according to the type of approach that clustering takes [5]. For example, CLIQUE uses a bottom-up search based on the Apriori association rule mining algorithm [11] that also uses a grid-based approach [12] where a subset of the dimensions forming small clusters are discovered first and are then expanded to find larger clusters with more attributes. PROCLUS uses a top-down approach starting with a predefined number of clusters from the full-dimensional space that are reduced to lower dimensions. Top-down approaches are capable of discovering subspaces, but less likely to discover projections [5].

Subspace clustering on categorical datasets has been performed by CLICKS [13]. CLICKS requires all objects in the same cluster to have the same value for each attribute in the subspace, which differs from P3C that allows multiple values in an interval of categorical values. CLICKS does not use the Apriori-like search, instead it performs a depth-first search for maximal cliques in a k-partite graph. HSM [14] performs density-based subspace clustering on heterogeneous data. Similarly to P3C, HSM can operate on datasets containing both numeric and categorical attributes. A projected clustering method was proposed by [15] for categorical datasets. [16] defined a dissimilarity measure for categorical data that gives more importance for some attributes in one cluster and more importance to other attributes in other clusters.

Existing algorithms for clustering have been modified for discovering subspaces with fuzzy approaches. The k-means approach was adapted [17] to include a fuzzy weighting of attributes of a cluster in a similar manner to LAC. Gustafson-Kessel clustering has been used for fuzzy attribute weighting [18] and possibilistic attribute weighting [19] in subspaces. Fuzzy c-means has been modified for fuzzy weighting of descriptors of subspace clusters [20]. Fuzzy or weighting approaches for subspace clustering of categorical data exist [21] [22], however, they are top-down approaches that require a predetermined number of clusters.

III. Preliminaries

Preliminary definitions are introduced before they are used in the main algorithm presented in Section IV. An overview of P3C is also presented before the extensions in 2P4C are presented in Section IV.

A. Definitions

These definitions follow on from the definitions in Section I.

For a categorical attribute a_j in 2P4C, a discrete possibility distribution of unordered categorical values represents an interval[1] Π on attribute a_j that is defined as

$$\Pi = \left\{ \frac{x_1}{\pi(x_1)}, \frac{x_2}{\pi(x_2)}, \ldots, \frac{x_h}{\pi(x_h)} \right\}, \tag{1}$$

where x is one of h values, and π is degree of membership. The possibility distribution of (1) extends P3C's

[1] An *interval* is different to a *fuzzy interval* (or *possibility interval*), which is characterised by a trapezoidal fuzzy set where $b = c$, or an interval-value fuzzy set.

crisp approach by allowing membership values with different degrees. An example of a categorical possibility distribution is presented in the results (15).

The width of interval Π is

$$\text{width}(\Pi) = \frac{\sum_{\pi \in (\Pi)} \pi}{\mid \text{dom}(a_j) \mid}, \tag{2}$$

which effectively normalises categorical data in a manner similar to P3C's normalisation of numerical data.

Let Π be an interval on attribute a_j. The support set of Π is $\text{SuppSet}(\Pi) = \{x \in D \mid \pi_\Pi(x^{(a_j)}) > 0\}$, which contains the data objects that belong to Π. The possibilistic support of Π is defined as

$$\text{PossSupp}(\Pi) = \sum_{x \in \text{SuppSet}(\Pi)} \pi_\Pi(x), \tag{3}$$

where π is the membership function of a possibility distribution Π for element x.

A possibilistic projected cluster has a *possibilistic p-signature* $\boldsymbol{P}$ that is defined as a set $\boldsymbol{P} = \{\Pi_1, \ldots, \Pi_p\}$ of p intervals on a (sub)set of p *distinct* attributes $\{a_{j_1}, \ldots, a_{j_p}\}(j_i \in \{1, \ldots, d\})$. For example, a possibilistic *3*-signature $\boldsymbol{P}$ from Table I might be

$$\boldsymbol{P} = \{\Pi_{\text{municipality}}, \Pi_{\text{music}}, \Pi_{\text{jobs}}\}, \tag{4}$$

where

$$\Pi_{\text{municipality}} = \left\{ \frac{\text{Beijing}}{\pi(\text{Beijing})}, \frac{\text{Bristol}}{\pi(\text{Bristol})}, \frac{\text{London}}{\pi(\text{London})} \right\}, \tag{5}$$

$$\Pi_{\text{music}} = \left\{ \frac{\text{hip hop}}{\pi(\text{hip hop})} \right\}, \tag{6}$$

$$\Pi_{\text{job}} = \left\{ \frac{\text{chef}}{\pi(\text{chef})} \right\}. \tag{7}$$

The support set of a possibilistic p-signature $\boldsymbol{P} = \{\Pi_1, \ldots, \Pi_p\}$ is defined as $\text{SuppSet}(\boldsymbol{P}) = \{x \in D \mid x \in \bigcap_{i=1}^{p} \text{SuppSet}(\Pi_i)\}$. The possibilistic support of a possibilistic p-signature $\boldsymbol{P}$ is

$$\text{PossSupp}(\boldsymbol{P}) = \sum_{x \in \text{SuppSet}(\boldsymbol{P})} \min_{\Pi \in \boldsymbol{P}} \pi_\Pi(x), \tag{8}$$

where the minimum is used for intersection of intervals in $\boldsymbol{P}$.

The concept of a possibilistic p-signature is used later in the 2P4C algorithm as candidate patterns for approximating a *true* possibilistic p-signature of a projected cluster. A *true* possibilistic p-signature $\tilde{\boldsymbol{P}}$ of a projected cluster (X_i, Y_i), $Y_i = \{a_1, \ldots, a_p\}$, is a possibilistic p-signature $\{\Pi_1, \ldots, \Pi_p\}$ where Π_i is the smallest interval on attribute a_i that projects the data points in X_i onto a_i.

B. The P3C algorithm

Pseudocode of the original P3C algorithm [3] is given below.

1) Discretise each attribute into bins.
2) Determine attributes with non-uniform distribution, and compute intervals that approximate projections of clusters onto these attributes.
3) Aggregate the intervals into cluster cores.
4) Refine cluster cores into projected clusters, compute outliers, and detect clusters' relevant attributes.

The first step of P3C is only relevant to numeric data, not categorical data. The second step finds "adjacency" relationships between bins, and uses these to create intervals of bins. The third step refers to using the intervals in a bottom-up search of clusters cores. The fourth step finds the membership matrix of data objects to projected clusters. 2P4C follows the same steps as P3C.

IV. The 2P4C Algorithm

2P4C uses the same algorithm as P3C (see Section III-B) combined with possibility distributions. Steps 2 and 3 of the P3C pseudocode were modified to incorporate the possibility distributions, however, the underlying principal of the P3C algorithm is the same.

The changes to step 2 are presented in Section IV-A. The changes include creating: intervals of items with possibility membership instead of crisp membership; measures of observed and expected possibilistic support instead of crisp support; and a continuous Poisson distribution instead of a discrete Poisson distribution.

The changes to step 3 are presented in Section IV-B. The changes include incorporating the measures of observed and expected possibilistic support into the same bottom-up search.

A. Approximating true possibilistic p-signatures

2P4C finds intervals—discrete possibility distributions of categorical values—that give a good approximation of true possibilistic *p*-signatures. This includes finding bins of categorical data that form the intervals.

1) Binning: The same approach as P3C for finding bins of categorical data is used here (step 1 of pseudocode in Section III-B). The idea is to find attribute bins with unusually high support that do not belong to a normal distribution. Attribute bins with non-uniform distribution may be relevant to a projected cluster, as defined in Section I.

There is a bin for every value in the domain of a categorical attribute. The support of every bin is calculated. The Chi-square test statistic is calculated with a confidence level fixed at $\alpha = 0.001$. The Chi-square statistic tests whether the bins in an attribute have a normal distribution. If the Chi-squared test determines an attribute to be non-uniform then the bin with the largest support is *marked*, and the remaining *unmarked* bins are tested again with the Chi-square test statistic. The process is repeated by marking the bin with the next largest support until the Chi-square test

statistic determines the *unmarked* bins of an attribute to have a normal distribution.

Now that the non-uniform bins are identified, intervals of marked bins are then created.

2) Intervals—Creating "adjacency": As previously mentioned in the introduction, an interval of numeric data is created based on "adjacency" in the natural order of real values. However, there are no consecutive values and no natural order with categorical data. Instead, an alternative approach for defining "adjacency" was created in P3C, and 2P4C improves the approach by better utilising the available data. We first describe our extension to P3C's Poisson-based criterion for creating and searching for intervals. We then introduce how the intervals are formed for discrete possibility distributions of categorical values.

For the continuous Poisson-based criterion, consider $\boldsymbol{P}$ to be a possibilistic p-signature, Π' to be an interval, and $\boldsymbol{R} = \boldsymbol{P} \cup \{\Pi'\}$ to be a larger possibilistic (p+1)-signature. For searching for possibilistic p-signatures, $\boldsymbol{R}$ is a candidate possibilistic (p+1)-signature to be tested; and for creating intervals, $\boldsymbol{R}$ is a combination of bins from two distinct attributes, which determines if they belong to the same cluster projection onto those attributes. The Poisson-based criterion assesses how likely the observed possibilistic support $\mathrm{PossSupp}(\boldsymbol{R})$ of $\boldsymbol{R}$ is with respect to the expected possibilistic support $\mathrm{EPossSupp}(\boldsymbol{R} = \boldsymbol{P} \cup \Pi')$ of $\boldsymbol{R}$ given $\boldsymbol{P}$. If the observed support is less likely then this provides strong evidence that Π' represents the same cluster as that of $\boldsymbol{P}$. The expected possibilistic support $\mathrm{EPossSupp}(\boldsymbol{R} = \boldsymbol{P} \cup \{\Pi'\})$ of $\boldsymbol{R}$ given $\boldsymbol{P}$ is defined as

$$\mathrm{EPossSupp}(\boldsymbol{R} = \boldsymbol{P} \cup \{\Pi'\} \mid \boldsymbol{P}) = \mathrm{PossSupp}(\boldsymbol{P}) \times \mathrm{width}(\Pi'). \quad (9)$$

The continuous Poisson-based criterion in 2P4C determines when the observed possibilistic support is *significantly larger* than the expected possibilistic support. A continuous extension $\mathrm{CPoisson}_\lambda(k)$ of the Poisson distribution handles real numbers of possibilistic support and is defined as

$$\mathrm{CPoisson}_\lambda(k) = \frac{\lambda^k e^{-\lambda}}{\Gamma(k)}, \quad (10)$$

where λ is a positive real number, which is equivalent to the expected possibilistic support, and k is the observed possibilistic support. The gamma function Γ substitutes the (discrete) Poisson distribution's denominator $k!$ and is defined as

$$\Gamma(k) = \int_0^\infty x^{k-1} e^{-x}\, \mathrm{d}x. \quad (11)$$

Extending the Poisson-based criterion to continuous values allows possibilistic and fuzzy events to be included, which relates to previous work on probability measures of fuzzy events [23].

For creating intervals, the Poisson-based criterion determines if two bins from two distinct categorical attributes belong to same cluster projection. Formally, according to P3C, consider two marked bins mb_1 and mb_2 from two distinct categorical attributes a_i and a_j ($i \neq j$), and a Poisson threshold P. Marked bins mb_1 and mb_2 belong to the same true possibilistic p-signature onto a_i and a_j if the following two conditions are satisfied.

1) $\mathrm{PossSupp}(\{mb_1\} \cup \{mb_2\}) >$
   $\mathrm{EPossSupp}(\{mb_1\} \cup \{mb_2\} \mid \{mb_1\})$, and
   $\mathrm{CPoisson}(\mathrm{PossSupp}(\{mb_1\} \cup \{mb_2\}),$
   $\mathrm{EPossSupp}(\{mb_1\} \cup \{mb_2\} \mid \{mb_1\})) < P$
2) $\mathrm{PossSupp}(\{mb_2\} \cup \{mb_1\}) >$
   $\mathrm{EPossSupp}(\{mb_2\} \cup \{mb_1\} \mid \{mb_2\})$, and
   $\mathrm{CPoisson}(\mathrm{PossSupp}(\{mb_2\} \cup \{mb_1\}),$
   $\mathrm{EPossSupp}(\{mb_2\} \cup \{mb_1\} \mid \{mb_2\})) < P$

P3C's "adjacency" relationship is based on transitive relations between all pairs of bins from distinct categorical attributes. If two marked bins mb_1 and mb_2 from the *same* categorical attribute a_i share at least one attribute from *another* categorical attribute a_j, ($i \neq j$), e.g., (mb_1, mb_3) and (mb_2, mb_3), then mb_1 and mb_2 are "adjacent". Intervals are formed from bins on the same attribute that have connected components of "adjacent" bins, which are crisp sets. P3C's requirement of sharing *at least* one attribute disregards the available information. For example, one shared attribute has the same meaning as ten shared attributes. So, 2P4C creates intervals that are possibility distributions to express different membership of bins according to the uncertainty and imprecision of the "adjacency" relationship.

3) Intervals—Discrete Possibility Distributions of Categorical Values: Both 2P4C and P3C use the same principal for binning data and defining "adjacency". 2P4C extends the principal of the Poisson-based criterion to handle continuous numbers for observed possibilistic support and expected possibilistic support. We now introduce how 2P4C determines membership of the intervals from "adjacency". Determining the possibilistic membership grades of discrete categorical data is based on a normalisation method [24].

Possibility distributions are suitable for modelling classes with non-sharp boundaries and gradual memberships that relate to uncertainty [25]. We adopt a possibility distribution [26] to also model the imprecision of the available information, because the connected components in an "adjacency" relationship provide a set of possible values and we can not fix a specific value, hence the set is considered to be imprecise. The possibility distribution models the uncertainty of the "adjacency" relationships between set members, which is considered to be unreliable or indeterministic because *a*) the co-occurrence of some paired set members may be more than that of other paired set members, and *b*) co-occurrences may be coincidental rather than relational. A probability distribution would be suitable if there was uncertainty and precision, however, this problem has uncertainty and imprecision [27].

We evaluate the relationships between values of the same attribute whilst others have learnt possibilistic graphical models of relations between different attributes [28].

For each variable, there is a contingency table with marked bins from the variable as rows and marked bins from all other attributes as columns. The crisp support of bin-bin pairs are the values in the table. For example, Table II is the contingency table of variable municipality in Table I.

TABLE II: Example contingency table

	music		job		
municipality	hip hop	reggae	chef	nurse	researcher
Beijing	1	0	1	0	0
Bristol	2	0	0	2	0
London	1	0	0	1	0
Ipswich	0	1	0	0	1

We use a frequency-based possibilistic approach for measuring membership grades. The possibilistic membership grade for member x_i in interval Π with $|\Pi| > 1$ is defined as

$$\pi_\Pi(x_i) = \frac{\sum_{j=1}^{|\boldsymbol{P'}|} \begin{cases} \frac{M_{ij}}{\sum_{k=1}^{|\Pi|} M_{kj}} & \text{if } \mathrm{QtyBins}(\Pi, j) > 1; \\ 0 & \text{otherwise;} \end{cases}}{|\boldsymbol{P'}|} \quad (12)$$

where $\boldsymbol{P'}$ is all other attributes and bins (e.g., {hip hop, reggae, chef, nurse, researcher} from Table II), and M is the contingency table with rows denoted as i and columns denoted as j. $\mathrm{QtyBins}$ determines the number of bins in an interval that share a value from another attribute. For example, $\mathrm{QtyBins}(\Pi_1, 1)$ for the example interval Π_1 and the 'hip hop' attribute of music gives 3 shared values (1, 2, and 1 in column hip hop of Table II). $\mathrm{QtyBins}$ is defined as

$$\mathrm{QtyBins}(\Pi, j) = \sum_{i=1}^{|\Pi|} \begin{cases} 1 & \text{if } M_{ij} > 0; \\ 0 & \text{otherwise.} \end{cases} \quad (13)$$

For the case where member x_i in possibilistic interval Π has the smallest possible value, i.e., $|\Pi| = 1$, the membership grade is defined as

$$\pi_\Pi(x_i) = \frac{1}{|\boldsymbol{P'}|}. \quad (14)$$

For the example in Table II, consider two intervals for municipality were created from the two conditions using the Poisson-based criterion in the previous stage. They are $\Pi_1 = \{\text{Beijing}, \text{Bristol}, \text{London}\}$ and $\Pi_2 = \{\text{Ipswich}\}$. The possibility distribution Π_1 is defined as

$$\begin{aligned} \Pi_1 &= \left\{ \frac{\text{Beijing}}{\frac{1/4}{5}}, \frac{\text{Bristol}}{\frac{2/4+2/3}{5}}, \frac{\text{London}}{\frac{1/4+1/3}{5}} \right\}, \\ &= \left\{ \frac{\text{Beijing}}{.05}, \frac{\text{Bristol}}{0.2\dot{3}}, \frac{\text{London}}{0.11\dot{6}} \right\}. \end{aligned} \quad (15)$$

A normal possibility distribution Π has a maximum membership value of 1, i.e., $\exists x'$ such that $\pi_A(x') = 1$. However, our proposed method relaxes the requirement for normality, so the membership values for a discrete categorical possibility distribution Π can be sub-normal, i.e., $\nexists x'$ such that $\pi_\Pi(x') = 1$. The reason for relaxing this requirement is to allow a weighting of set members relative to each other. This is a similar reason to weighting linguistic variables modelled with fuzzy sets in medical applications [29]. Sub-normal possibility distributions cause issues concerning the necessity measure that are addressed by the certainty measure [30].

B. Cluster Core Search

Once true possibilistic p-signatures are approximated, an Apriori-like search for cluster cores is performed. 2P4C extends the method from P3C to handle possibility distributions.

A possibilistic p-signature represents a projected cluster core C if $\boldsymbol{P}$ consists of *1*) *only* and *2*) *all* intervals that represent cluster C. So, a possibilistic p-signature $\boldsymbol{P} = \{\Pi_1, \ldots, \Pi_p\}$ with support set $\mathrm{SuppSet}(\boldsymbol{P})$ is a cluster core if:

1) For any possibilistic q-signature $\boldsymbol{Q} \subseteq \boldsymbol{P}$ where $q = 1, \ldots, p-1$ and interval $\Pi' \in \boldsymbol{P} \setminus \boldsymbol{Q}$, it holds that:
   $\mathrm{PossSupp}(\boldsymbol{Q} \cup \{\Pi'\}) > \mathrm{EPossSupp}(\boldsymbol{Q} \cup \{\Pi'\} \mid \boldsymbol{Q})$, and
   $\mathrm{CPoisson}(\mathrm{PossSupp}(\boldsymbol{Q} \cup \{\Pi'\}), \mathrm{EPossSupp}(\boldsymbol{Q} \cup \{\Pi'\} \mid \boldsymbol{Q}) < P$
2) For any interval Π' not in $\boldsymbol{P}$, it holds that:
   $\mathrm{PossSupp}(\boldsymbol{P} \cup \{\Pi'\}) \leq \mathrm{EPossSupp}(\boldsymbol{P} \cup \{\Pi'\} \mid \boldsymbol{P})$, and
   $\mathrm{CPoisson}(\mathrm{PossSupp}(\boldsymbol{P} \cup \{\Pi'\}), \mathrm{EPossSupp}(\boldsymbol{P} \cup \{\Pi'\} \mid \boldsymbol{P}) \geq P$

The first condition satisfies the downward closure property, which means for any possibilistic p-signature $\boldsymbol{P}$ that satisfies the first condition, any sub-signature of $\boldsymbol{P}$ also satisfies the first condition. This property allows an Apriori-like search.

The second condition satisfies the conditions for a possibilistic p-signature to be maximal, which means for any possibilistic p-signature $\boldsymbol{P}$ that satisfies the first condition, there are no super-signatures of $\boldsymbol{P}$.

Projected clusters are computed from the cluster cores and support sets according to P3C, which also detects outliers that do not belong to a projected cluster.

V. Experimentation

2P4C and P3C were compared on multiple real-world datasets. The aims of the experiments were *a*) to analyse performance of the two algorithms and *b*) to analyse the benefit of using possibility distributions. The datasets are discussed first followed by experimentation.

All runs of both algorithms were performed with a Poisson threshold of $1.0\mathrm{e}^{-10}$. The experiments were conducted on a Windows 7 PC with a 2.4 GHz CPU and 4 GB RAM.

A. Datasets

Six real-world benchmark datasets from the University of California, Irvine (UCI)[2] repository were used. The datasets are listed in Table III with their properties.

It has been shown that as the number of dimensions increases in different runs of (full-space) clustering, the similarity between data objects/records starts to degrade at 10 dimensions [1]. So, a criterion for choosing datasets was to have different size dimensions. The number of data objects/records was also varied. All selected datasets are labelled, so the clustering accuracy can be measured and comparisons made between the two algorithms, which is discussed in the next section. The Hepatitis and Molecular biology (promoter gene sequences) datasets were not used because preliminary experiments did not discover projected clusters.

TABLE III: Properties of UCI datasets

Dataset	# Records	# Categorical attributes
Breast Cancer Wisconsin	699	10
Congressional Voting	435	16
Lymphography	148	18
Mushroom	8124	22
Soybean	307	35
Splice[3]	3190	61

B. Performance analysis

The performance of 2P4C and P3C were compared using the following measures.

- *Number of clusters*.
- *Execution time* of the algorithm.
- *Distribution of cluster size* measures the percentage of clusters that have few or many attributes. For example, the number of clusters with two dimensions, the number of clusters with three dimensions, and so on.
- *Average support size per cluster size* measures the average support for each (small/large) cluster size. For example, the average support for clusters with two dimensions, the average support for clusters with three dimensions, and so on.
- F_1 *score* measures clustering accuracy. The F_1 score is often used to measure supervised learning methods, however, it is used here on unsupervised learning methods with labelled data to measure the accuracy of finding real clusters in the datasets and to compare accuracy between 2P4C and P3C. For each projected cluster i, the true cluster j^i (from labelled dataset) with the largest number of shared data points is determined. Precision is calculated for projected cluster i as the number of data points shared with projected cluster i and true cluster j^i divided by the total number of data points in i. Recall is calculated for projected cluster i as the number of data points shared with projected cluster i and true cluster j^i divided by the total number of data points in j^i. The F_1 score of projected cluster i is the harmonic mean of precision and recall, which is defined as

$$F_1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}. \quad (16)$$

The F_1 score of a clustering algorithm is the mean of all projected cluster F_1 scores.

The number of clusters and F_1 scores are presented in Table IV. 2P4C discovered less clusters than P3C in three of the datasets. One of those datasets, Splice, was terminated after running for many hours because the number of candidate p-signatures was increasing at a high rate (almost by a factor of one) at each p, which means a very large number of clusters would have been produced after a very large execution time. 2P4C discovered more clusters than P3C in one of the datasets. In the other two datasets, P3C discovered no clusters in one dataset and 2P4C discovered no clusters in the other.

Three results do not contain F_1 scores for either P3C or 2P4C because no clusters were discovered or P3C was terminated early. For the other three datasets, the F_1 scores were larger for 2P4C than P3C in two datasets and smaller in one dataset. This suggests 2P4C is able to discover more accurate projected clusters than P3C on most of the datasets used.

TABLE IV: Performance results

Dataset	# Clusters		F_1 score	
	P3C	2P4C	P3C	2P4C
Breast Cancer Wisconsin	29	32	0.62	0.73
Congressional Voting	0	7	0	0.64
Lymphography	1	0	0.72	0
Mushroom	71	23	0.60	0.63
Soybean	59	2	0.40	0.27
Splice	—	8	—	0.34

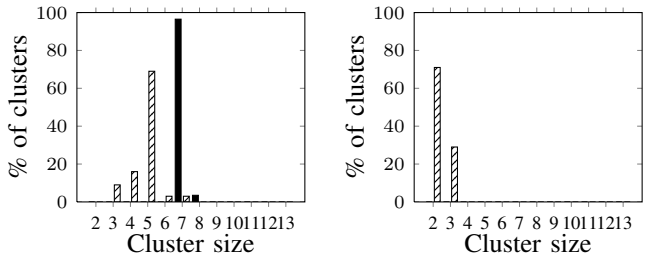
The percentage of clusters with different numbers of attributes (cluster size) is presented in Figure 1. The purpose is to explore the cluster size between both algorithms. For the datasets where clusters were discovered by both algorithms (Figures 1a, 1d, and 1e), 2P4C discovered more clusters that were smaller than P3C. For the two datasets where 2P4C produced clusters but P3C did not (Figures 1b and 1f), 2P4C discovered clusters that were small, i.e., containing two or three attributes. This demonstrates that 2P4C produces more smaller clusters than 2P4C.

Figure 2 shows the average support per cluster size. The support values from 2P4C are lower than P3C, which was expected, because 2P4C handles partial membership with possibility distributions whilst P3C has Boolean membership. Some of the possibilistic supports are very small, particularly in Figures 2d and 2f. An observation is that the average supports have similar magnitudes per cluster size within each dataset.

The execution time of 2P4C, in Table V, was less than P3C in all datasets. A possible reason is that fewer clusters

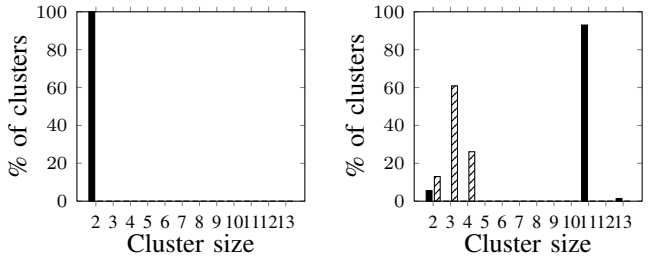
[2]http://archive.ics.uci.edu/ml/

[3]Full name is Molecular biology (splice-junction gene sequences)



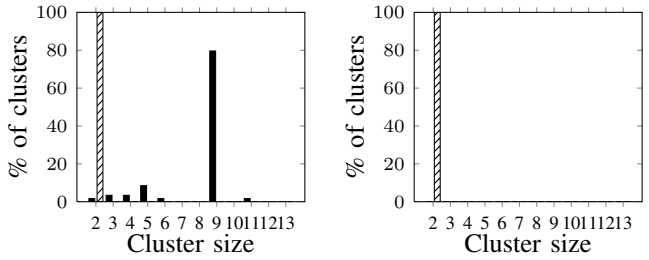
(a) Breast Cancer Wisconsin

(b) Congressional Voting



(c) Lymphography

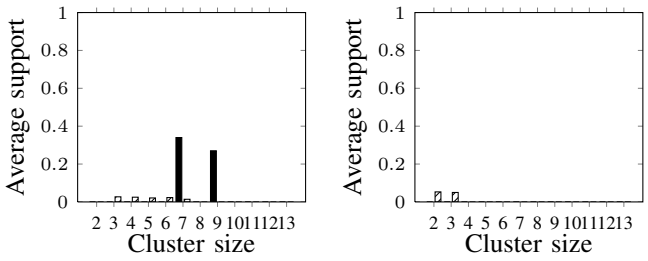
(d) Mushroom



(e) Soybean

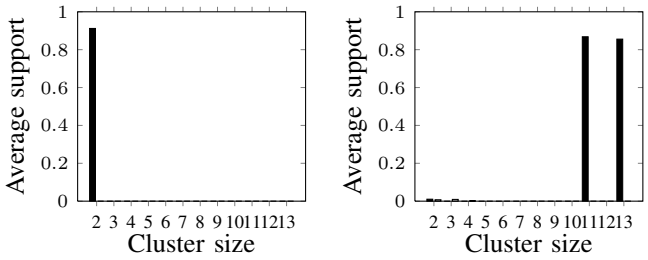
(f) Splice

Fig. 1: Distribution of cluster size for each dataset. Black is P3C, diagonal pattern is 2P4C.



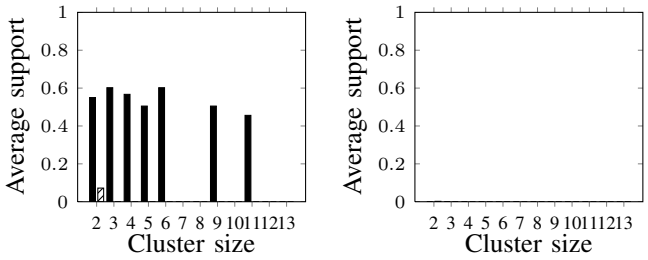
(a) Breast Cancer Wisconsin

(b) Congressional Voting



(c) Lymphography

(d) Mushroom



(e) Soybean

(f) Splice

Fig. 2: Average support per cluster size for each dataset. Normalised within $[0, 1]$. Black is P3C, diagonal pattern is 2P4C.

were discovered and the cluster sizes were smaller in 2P4C. Larger execution times are observed for the Mushroom and Splice datasets, which both have a large number of records and attributes. No execution time of P3C was recorded for the Splice dataset because it was terminated during execution for the previously stated reason.

TABLE V: Execution times

Dataset	Execution time (s)	
	P3C	2P4C
Breast Cancer Wisconsin	2.76	1.97
Congressional Voting	0.72	0.65
Lymphography	1.19	1.08
Mushroom	6853.50	77.31
Soybean	6.20	1.80
Splice	—	67.01

C. Analysis of nested clusters

In the previous section, Figure 1 showed that 2P4C produced projected clusters with fewer attributes than P3C. To identify whether the algorithms discovered similar or different clusters, an analysis was performed on the number of clusters from one algorithm that were nested inside clusters from the other algorithm. A nested projected cluster would suggest a more compact cluster. Nested means the entire possibilistic p-signature of one cluster is contained with a cluster from the other algorithm. For example, the possibilistic p-signature

$$\{\text{municipality} = \{\text{Beijing}, \text{Bristol}, \text{London}\}, \\ \text{music} = \{\text{hip hop}\}\},$$

is nested within possibilistic p-signature

$$\{\text{municipality} = \{\text{Beijing}, \text{Bristol}, \text{London}\}, \\ \text{music} = \{\text{hip hop}\}, \\ \text{job} = \{\text{chef}\}\},$$

because all of the intervals from the first projected cluster—municipality and music—are contained within the second projected cluster.

The projected clusters produced from 2P4C and P3C were compared for all datasets, and the results are reported in Table VI. A "—" indicates no result, because there were

no cluster results from either P3C or 2P4C to allow a comparison. 2P4C contained more nested clusters than P3C in two of the datasets, and there were no nested clusters in third dataset, Soybean. Note that 2P4C produced few clusters from the Soybean dataset, which may account for no nested clusters. The observation that 2P4C produces smaller clusters with some of the same attributes as those clusters from P3C could further support the suggestion that 2P4C discovers compact clusters (fewer attributes) when compared to P3C.

TABLE VI: Number of nested clusters

Dataset	# P3C clusters nested in 2P4C clusters	# 2P4C clusters nested in P3C clusters
Breast Cancer Wisconsin	0	245
Congressional Voting	—	—
Lymphography	—	—
Mushroom	3	92
Soybean	0	0
Splice	—	—

VI. Conclusions

This paper demonstrates the viability and potential benefits of adding variable membership grades with possibility distributions in 2P4C. 2P4C produced higher clustering accuracy and fewer clusters in most of the datasets. On one dataset, P3C discovered no clusters whilst P3C discovered multiple clusters. The number of clusters per cluster size and the number of nested clusters suggests that 2P4C has more compact clusters.

This paper's contribution is that fewer, more accurate, more compact, and new clusters can be discovered with 2P4C when compared to P3C. To understand any importance of this, the meaning of the clusters will be analysed in future work with a real-world application. Scalability analysis will also explore potential limitations of P3C that 2P4C might overcome, such as the observed large execution times of P3C. Alternative representations and learning methods of variable membership grades may also be explored.

References

[1] K. Beyer, J. Goldstein, R. Ramakrishnan, and U. Shaft, "When is "nearest neighbor" meaningful?" in *Database Theory – ICDT99*, ser. Lecture Notes in Computer Science, C. Beeri and P. Buneman, Eds., vol. 1540. Springer Berlin Heidelberg, 1999, pp. 217–235.

[2] A. Zimek, "Clustering high-dimensional data," in *Data Clustering: Algorithms and Applications*, C. C. Aggarwal and C. K. Reddy, Eds. CRC Press, 2013.

[3] G. Moise, J. Sander, and M. Ester, "Robust projected clustering," *Knowledge and Information Systems*, vol. 14, no. 3, pp. 273–298, 2008.

[4] W. W. Cohen, "Learning trees and rules with set-valued features," in *Proceedings of the Thirteenth National Conference on Artificial Intelligence - Volume 1*, ser. AAAI'96, 1996, pp. 709–716.

[5] H.-P. Kriegel, P. Kröger, and A. Zimek, "Clustering high-dimensional data: A survey on subspace clustering, pattern-based clustering, and correlation clustering," *ACM Transactions on Knowledge Discovery from Data*, vol. 3, no. 1, pp. 1:1–1:58, Mar. 2009.

[6] R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan, "Automatic subspace clustering of high dimensional data for data mining applications," in *Proceedings of the 1998 ACM SIGMOD*, 1998, pp. 94–105.

[7] C. C. Aggarwal, J. L. Wolf, P. S. Yu, C. Procopiuc, and J. S. Park, "Fast algorithms for projected clustering," in *Proceedings of the 1999 ACM SIGMOD*, 1999, pp. 61–72.

[8] C. Domeniconi, D. Gunopulos, S. Ma, B. Yan, M. Al-Razgan, and D. Papadopoulos, "Locally adaptive metrics for clustering high dimensional data," *Data Mining and Knowledge Discovery*, vol. 14, no. 1, pp. 63–97, 2007.

[9] T. Hofmann, "Probabilistic latent semantic indexing," in *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR '99, 1999, pp. 50–57.

[10] C. C. Aggarwal, "On the equivalence of PLSI and projected clustering," *ACM SIGMOD Record*, vol. 41, no. 4, pp. 45–50, 2013.

[11] R. Agrawal and R. Srikant, "Fast algorithms for mining association rules in large databases," in *Proceedings of the 20th International Conference on VLDB*, San Francisco, CA, USA, 1994, pp. 487–499.

[12] W. Cheng, W. Wang, and S. Batista, "Grid-based clustering," in *Data Clustering: Algorithms and Applications*, C. C. Aggarwal and C. K. Reddy, Eds. CRC Press, 2013.

[13] M. J. Zaki, M. Peters, I. Assent, and T. Seidl, "CLICKS: An effective algorithm for mining subspace clusters in categorical datasets," in *Proceedings of the Eleventh ACM SIGKDD*, 2005, pp. 736–742.

[14] E. Müller, I. Assent, and T. Seidl, "HSM: Heterogeneous subspace mining in high dimensional data," in *Proc. 21st International Conference on Scientific and Statistical Database Management (SSDBM 2009)*, 2009, pp. 497–516.

[15] M. Kim and R. Ramakrishna, "Projected clustering for categorical datasets," *Pattern Recognition Letters*, vol. 27, no. 12, pp. 1405–1417, 2006.

[16] J. Lee and Y.-J. Lee, "An effective dissimilarity measure for clustering of high-dimensional categorical data," *Knowledge and Information Systems*, 2013.

[17] G. Gan, J. Wu, and Z. Yang, "A fuzzy subspace algorithm for clustering high dimensional data," in *Advanced Data Mining and Applications*, ser. Lecture Notes in Computer Science, X. Li, O. R. Zaïane, and Z.-h. Li, Eds., 2006, vol. 4093, pp. 271–278.

[18] C. Borgelt, *Fuzzy Subspace Clustering*. Springer Berlin Heidelberg, 2010, pp. 93–103.

[19] C. Puri, "Objective function based fuzzy subspace clustering," Ph.D. dissertation, University of Delhi, 2013.

[20] K. Simiński, "Clustering in fuzzy subspaces," *Theoretical and Applied Informatics*, vol. 24, no. 4, pp. 313–326, 2012.

[21] E. Y. Chan, W. K. Ching, M. K. Ng, and J. Z. Huang, "An optimization algorithm for clustering using weighted dissimilarity measures," *Pattern Recognition*, vol. 37, no. 5, pp. 943–952, 2004.

[22] F. Cao, J. Liang, D. Li, and X. Zhao, "A weighting k-modes algorithm for subspace clustering of categorical data," *Neurocomputing*, vol. 108, no. 0, pp. 23–30, 2013.

[23] L. Zadeh, "Probability measures of fuzzy events," *Journal of Mathematical Analysis and Applications*, vol. 23, no. 2, pp. 421–427, 1968.

[24] D. Nauck and R. Kruse, "Fuzzy classification rules using categorical and metric variables," in *Fuzzy-Neuro Systems 1999 - Computational Intelligence (FNS'99)*, G. Brewka, R. D. Gottwald, and A. Schierwagen, Eds. Leipzig: Leipziger Universitätsverlag, 1999, pp. 133–144.

[25] L. A. Zadeh, "Test-score semantics for natural languages and meaning representation via pruf," in *Fuzzy Sets, Fuzzy Logic, and Fuzzy Systems*, G. J. Klir and B. Yuan, Eds. River Edge, NJ, USA: World Scientific Publishing Co., Inc., 1996, pp. 542–586.

[26] L. Zadeh, "Fuzzy sets as a basis for a theory of possibility," *Fuzzy Sets and Systems*, vol. 1, no. 1, pp. 3–28, 1978.

[27] R. Kruse, C. Borgelt, and D. Nauck, "Problems and prospects in fuzzy data analysis," in *Intelligent Systems and Soft Computing: Prospects, Tools and Applications*, 2000, pp. 95–109.

[28] C. Borgelt and R. Kruse, "Learning possibilistic graphical models from data," *IEEE Transactions on Fuzzy Systems*, vol. 11, no. 2, pp. 159–172, 2003.

[29] J. Garibaldi and R. John, "Choosing membership functions of linguistic terms," in *The 12th IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2003)*, vol. 1, 2003, pp. 578–583.

[30] R. R. Yager, "A modification of the certainty measure to handle subnormal distributions," *Fuzzy Sets and Systems*, vol. 20, no. 3, pp. 317–324, 1986.